Grounding LLM Explanations for Process Mining via Agentic Function Calling

Oleksii Kalashnikov[1,2,*,†], Jonas Amling[1,*,†], Christian Dormagen[3], Andreas Fischer[2], Stephan Scheele[4,*] and Ute Schmid[3]

[1] *dab: Daten - Analysen & Beratung GmbH, Hans-Obser-Straße 12, 94469 Deggendorf, Germany*
[2] *Deggendorf Institute of Technology, Dieter-Görlitz-Platz 1, 94469 Deggendorf, Germany*
[3] *Otto-Friedrich-University Bamberg, Kapuzinerstraße 16, 96047 Bamberg, Germany*
[4] *OTH Regensburg University of Applied Sciences, Seybothstraße 2, 93053 Regensburg, Germany*

Abstract

Interpreting process mining outputs such as process-variant clusters and conformance violations is difficult for non-experts without clear explanations. While large language models (LLMs) can generate natural language narratives, they are unreliable when reasoning over numerical data and process structures. We present a demonstrator that constrains an LLM to an agentic function-calling interface grounded in precomputed XAI outputs, combining density-based clustering, rule-based explanations, and both alignment-based and declarative conformance checking. The system answers natural language queries by retrieving and composing computed metrics and explanations rather than inventing them. We evaluate the demonstrator on the Road Traffic Fine Management Process dataset across queries of increasing complexity.

Keywords

Process Mining, Conformance Checking, eXplainable AI, Agentic Explanations

1. Introduction

For data storytelling to be truly effective, it must be accompanied by clear narratives that guide users through the insights and implications of the data [1, 2]. Large Language Models (LLMs) provide a natural solution for generating such narratives, enabling dynamic, context-aware and user-specific explanations. [3, 4] Unfortunately, explanations directly generated by LLMs fail for complex compositional problems [5] and lack the mathematical reasoning capabilities required for reliable numerical analysis or ranking [6]. Moreover, they are often not grounded in the underlying process [7]. At best, this leads to "explanations without explainability" [8] and at its worst to misinformation and incorrect explanations [9]. Therefore, LLMs must be constrained [10] and only generate verbalizations directly from the underlying data or recall explanations derived from established explanation methods [11]. Crucially, in our approach, the LLM itself is not the underlying black-box model; instead, it serves as an interface that verbalizes clustering explanations and computed metrics retrieved via function calling. Their actual contribution is then reduced to translating the insights and generated explanation into a understandable natural language narrative that is more accessible to the target audience [12, 13, 14].

Process Mining (PM) aims to to analyze event data from operational processes to gain insights and improve their performance, compliance, and efficiency [15]. Two key challenges in process mining are improving usability and understandability for non-experts. Thus, results must be presented through intuitive interfaces and accompanied by clear, trustworthy explanations so that non-experts can correctly

Late-breaking work, Demos and Doctoral Consortium, colocated with the 4th World Conference on eXplainable Artificial Intelligence: July 01–03, 2026, Fortaleza, Brazil

*Corresponding author.
† These authors contributed equally.
oleksii.kalashnikov@stud.th-deg.de (O. Kalashnikov); jonas.amling@dab-gmbh.de (J. Amling); christian.dormagen@uni-bamberg.de (C. Dormagen); andreas.fischer@th-deg.de (A. Fischer); stephan.scheele@oth-regensburg.de (S. Scheele); ute.schmid@uni-bamberg.de (U. Schmid)
0000-0002-0877-7063 (O. Kalashnikov); 0009-0000-1554-9950 (J. Amling); 0009-0006-7744-746X (C. Dormagen); 0000-0002-0074-541 (A. Fischer); 0000-0003-0787-3181 (S. Scheele); 0000-0003-0787-3181 (U. Schmid)

interpret the insights [16, 17]. In reality, real word processes can lead to so called "spaghetti models" [15] and one way of untangling these models is by subprocess discovery via clustering[18]. Another key task in process mining is conformance checking [15], which evaluates how well observed process executions align with a reference process model. Two common approaches include alignment-based conformance checking [19], which detects deviations from procedural models using Petri nets, and DECLARE-based conformance checking [20, 21], which evaluates compliance against declarative rule-based models.

We present a PM demonstrator extending earlier work [18, 22]. While our previous XAI-2025 demonstrator, ClustXRAI [23], focused on interactive cluster exploration with static XAI models, this work adds alignment-based and declarative conformance checking for detecting process deviations, along with an autonomous agentic LLM that explains insights on demand via a grounded function-calling interface.It utilizes clustering to identify distinct execution patterns, thereby managing the complexity of real-world processes. Additionally, we integrate conformance checking to detect deviations between observed executions and normative process models. To support the interpretation of results, explainable artificial intelligence (XAI) methods are employed to characterize and explain the identified patterns. Finally, to make these technical insights accessible to non-experts, we integrate an agentic LLM as an interactive interface.It autonomously queries underlying metrics, clustering details, and explanations via function calling to generate dynamic, on-demand natural language narratives.

2. System Overview

Before demonstrating the system on a real-world process mining dataset, we first introduce its architecture and key components as the foundation for the walkthrough.

2.1. Architecture

As illustrated in Figure 1, our system architecture is divided into two distinct processing phases: a preprocessing pipeline and an agentic LLM interaction layer. The preprocessing pipeline begins with the extraction of raw event data from an ERP system, which is transformed into a structured event log. This log serves as the foundation for process discovery and clustering algorithms that identify distinct process variants. In the second component, the data undergoes conformance checking (alignment-based and declarative), where traces are evaluated against the normative process model. Finally, we generate explanations for both the resulting clusters and the conformance checking outcomes, incorporating background knowledge at each stage to enrich the analysis. These XAI explanations are available either precomputed or computed lazily on demand via function calls. The second component features an interactive dashboard accompanied by an agentic LLM. This agent enables a conversational interface where users can interactively explore the data. To answer user queries, the agent uses a set of tools to retrieve the necessary information from the data underlying the dashboard and the preprocessing pipeline, such as specific metrics and clustering or conformance explanations.

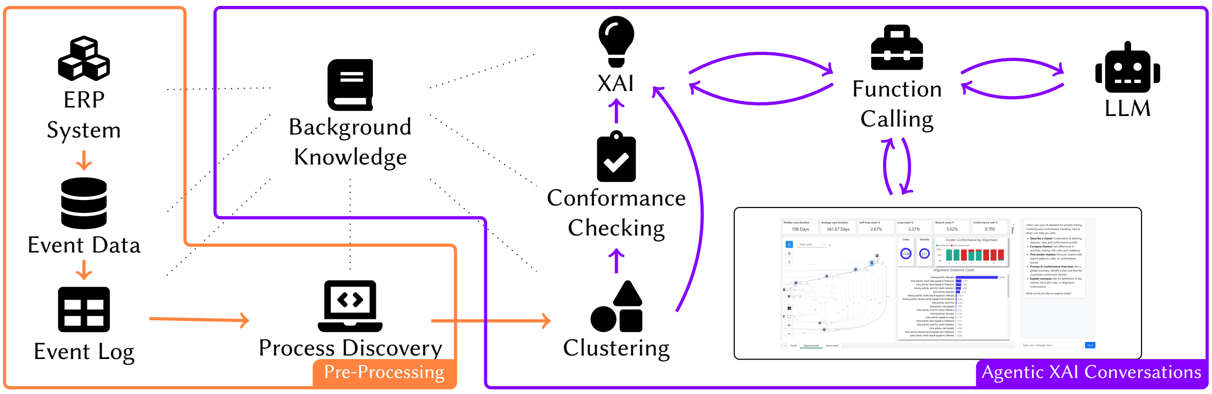

Figure 1: Architecture diagram showing *preprocessing stage* and user interaction with *agentic LLM explanations*

2.2. Components

Process Discovery For process discovery, we utilize the Microsoft Power Automate Process Mining solution[1]. The tool discovers a process model from the event data, represented as a Directly-Follows Graph (DFG), which reflects the underlying structure of the event log.

Clustering Our clustering approach follows previous work [18, 23]. We employ a density-based algorithm [24] combined with activity profiling and variant-based downsampling. Each case in the event log is mapped to its process variant and encoded as a boolean feature vector capturing trace-level constraints (e.g., activity presence and temporal ordering). To approximate the underlying case-level distribution, we sample representative variants and compute pairwise similarities via a distance matrix, to which the clustering algorithm is applied to identify clusters and filter noise. Finally, the resulting variant-level assignments are projected back onto the individual cases to enable detailed analysis.

Conformance Checking Conformance checking is performed through two complementary pipelines, both accepting a normative process model (in the form of Business Process Model and Notation – BPMN) and the clustered event log as input, producing an extended event log in which each trace is enriched with its detected violations. The first pipeline utilizes an alignment-based approach [19], where the BPMN model is automatically converted into a Petri net and optimal alignments are computed. Deviations are recorded as *log moves* (activities executed in a trace that the model does not permit) and *model moves* (activities required by the model but absent from the trace), which are appended to the extended log. The second pipeline adopts a rule-based approach grounded in Declare [21], where the BPMN model is automatically translated into a set of declarative constraints and evaluated against each trace. Violations are recorded as unsatisfied constraints, which are appended to the extended log.

XAI We combine two complementary explainers to characterize the discovered clusters. The first is a domain-agnostic explainer based on ClusterExplainR [25], which derives rule-based cluster descriptions by computing local feature importance scores regarding activity presence and absence. The second utilizes trace-level conformance violations, encompassing both alignment-based and declarative deviations as described in Section 2.2. These violations are aggregated at the cluster level to provide conformance rates and a ranked breakdown of specific violations.

Agentic Explanations via Function Calling Instead of relying on pre-cached text or retrieval-augmented generation, our approach provides the LLM with a suite of typed *function calls*. These structured interfaces allow the agent to autonomously retrieve, calculate, and compare process metrics and explanations on demand. When a user poses a query, the LLM identifies and executes the necessary functions, processes the returned data, and generates a natural language response. This agentic approach ensures that every answer is grounded in precise, computed values. In total, we implemented 16 functions that fall into four categories: retrieval, comparison, similarity, and context. Retrieval functions such as `get_cluster_conformance` return a cluster's full conformance profile (rate, case count, ranked violations) given its ID and conformance type. Comparison functions like `compare_clusters_conformance` contrast multiple clusters, detailing shared and unique violations. Equivalent functions compare activity presence, metrics, and rules. Similarity functions such as `get_most_similar_clusters_activity` identify the top-n clusters resembling a target via pairwise Euclidean distances over activity vectors or rule distributions. Context functions like `get_declare_rule_type_explanations` provide domain definitions for Declare constraints to ensure semantically grounded explanations.

Dashboard The dashboard shown in Figure 2 is implemented in Power BI and integrates interactive process visualizations, conformance metrics, page navigation, and an LLM-based chat interface.

[1]https://learn.microsoft.com/en-us/power-automate/process-mining-overview, 26 February 2026.

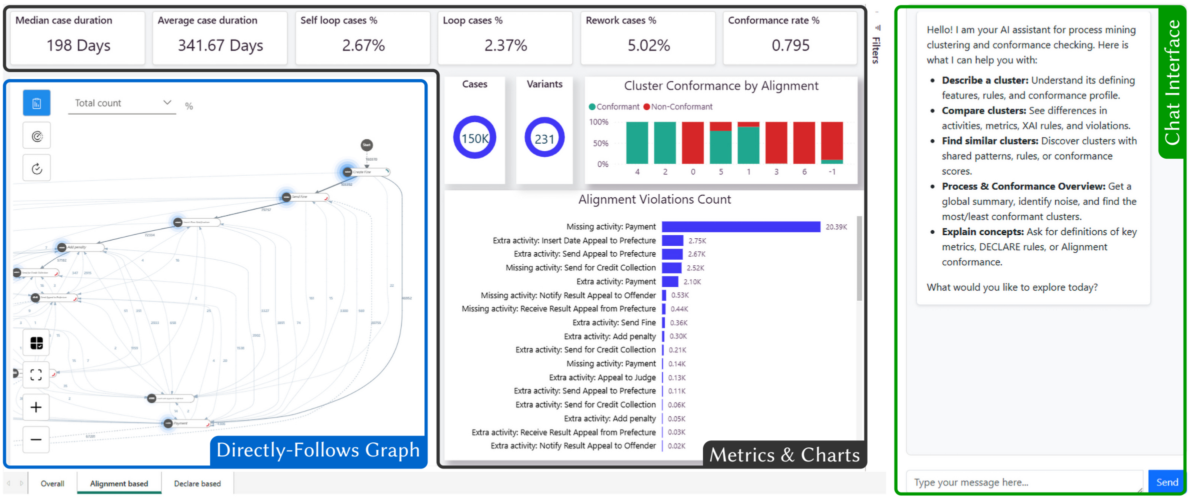

Figure 2: Interactive dashboard displaying the DFG visual, key metrics and conformance charts, page selection, and the agentic chat interface.

The overall process flow is represented by a Directly-Follows Graph (DFG), initially showing the full dataset with activity nodes and frequency-labeled edges. When clusters are selected, the graph filters to the relevant subset. Merging multiple clusters, as shown for Clusters 4 and 6 in Figure 3, allows analysts to compare subprocess structures and map alignment violations directly onto the process flow. The classical process flow performance indicators, including case duration statistics, rework rates (self-loops, loops), and the overall conformance rate, alongside indicators for case volume and variant count, provide an overall overview of the process. The "Cluster Conformance by Alignment" chart visualizes the ratio of conformant to non-conformant cases per cluster, while "Alignment Violations Count" ranks deviations by frequency (e.g., *Extra activity: Send for Credit Collection*). Selecting elements in these charts dynamically filters the entire dashboard, including the DFG. Navigation tabs for switching between the alignment-based view, the Declare-based dashboard, and the clustering overview, are provided on the bottom left. Finally, an LLM-based chat interface allows users to query conformance data using natural language, providing analytical insights through dialogue-driven, context-aware explanations. This interface explicitly displays the invoked function call stack alongside the response, enabling users to verify the correctness of the retrieved data when needed.

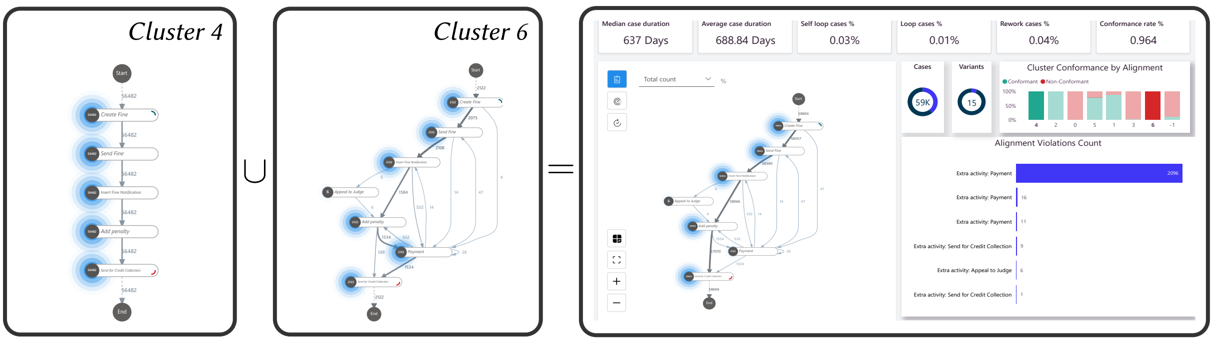

Figure 3: Dashboard displaying the combined Process Model of selected Clusters 4 and 6.

3. Walkthrough: Grounded LLM Explanations via Function Calling

We demonstrate our system using the Road Traffic Fine Management Process (RTFMP) dataset[2] derived from an Italian local police force information system. This dataset documents the full lifecycle of traffic fines, from issuance to settlement. The following scenarios show how the agentic LLM autonomously resolves user queries by invoking function calls. We progress from simple lookups to complex multi-step analyses, presenting the query, executed function calls, and the generated narrative for each case.

3.1. Single Cluster Conformance Analysis

In response to the query *"Explain cluster 6 in terms of declare conformance"*, the agent executes `get_cluster_conformance(cluster_id:"6", cc_type:"Declare")`. This call reveals that among 2,122 cases, none (0.0%) are conformant due to a single violation: Response(Payment, {Payment, END}). Interpreting this, the LLM states that every trace fails the constraint requiring a *Payment* to be followed by another *Payment* or process termination (Figure 4). The generated narrative follows the structured output of the underlying function call, making the response largely traceable to computed data.

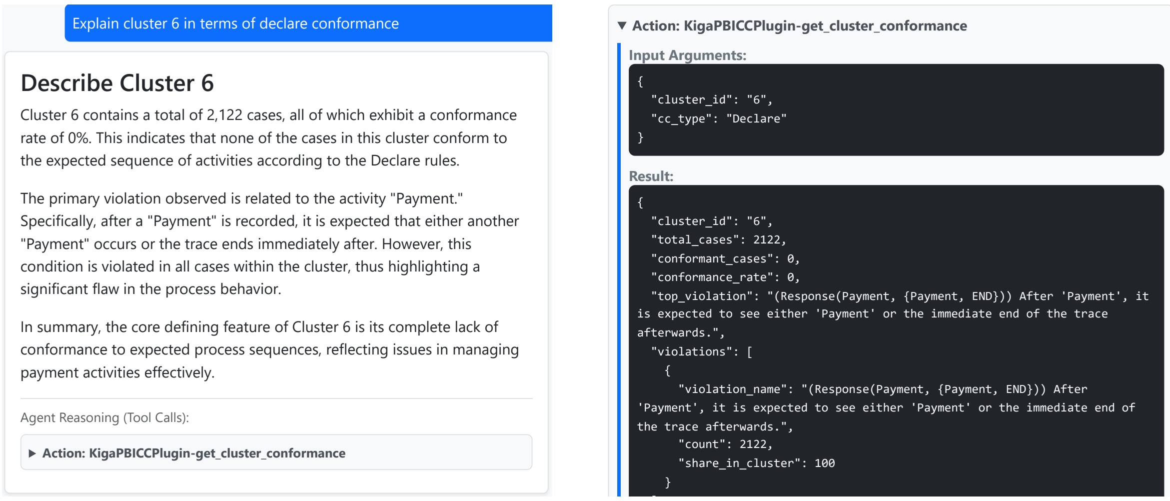

Figure 4: Cluster 6 conformance: LLM response (left) and tool call (right).

3.2. Direct Cluster Conformance Comparison

When asked to *"compare clusters 6 and 0 in terms of declarative conformance"*, the agent triggers function `compare_clusters_conformance(cluster_ids:["6","0"], cc_type:"Declare")`, which returns conformance rates along with lists of shared and unique violations. Specifically, Cluster 6 fails the *Payment* response constraint, whereas Cluster 0 violates two rules related to *Send Fine* sequencing and mandatory end activities. Finally, these findings are synthesized into a concise comparison, as shown in Figure 5. The response can be traced back to the returned function call result, supporting verification of the reported violations and conformance rates.

3.3. Multi-Step Query Orchestration

The agent's capacity for multi-step reasoning is demonstrated in Figure 6, where it autonomously decomposes a complex analytical task. The user prompts the system to identify the cluster most similar to the least conformant one, compare them across activities, rules, and violations, and finally evaluate them against the dataset average. First, `get_clusters_by_conformance_rate` and `get_similar_clusters_by_conformance` are used to identify Cluster 0 as the least conformant and Cluster 6 as its most similar counterpart. Next, `get_cluster_info`, `compare_cluster_activities`,

[2] https://data.4tu.nl/articles/dataset/Road_Traffic_Fine_Management_Process/12683249, accessed 9 March 2026.

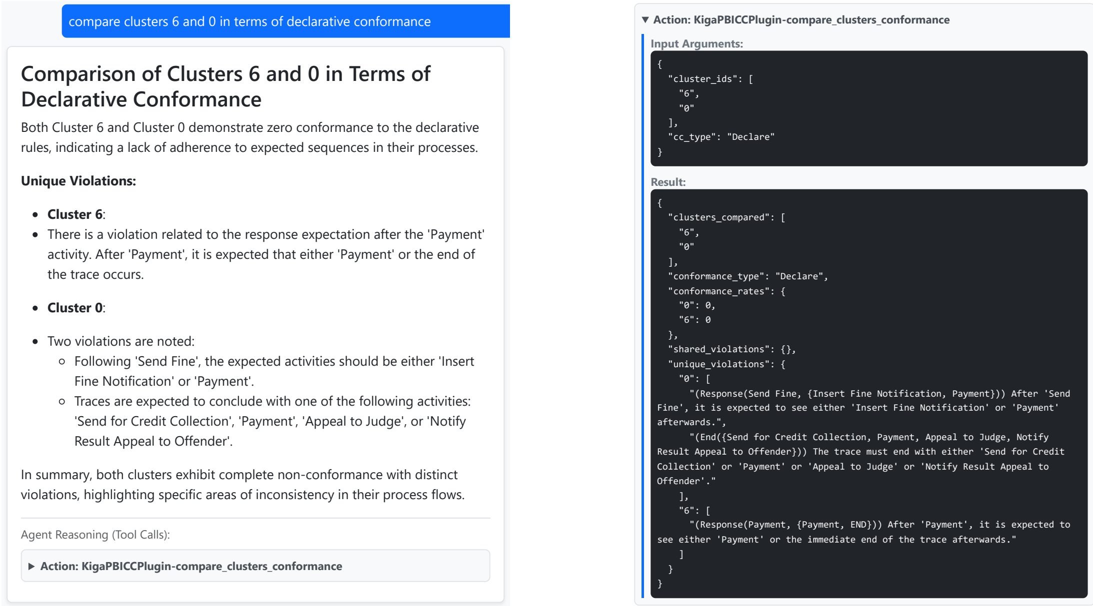

Figure 5: Cluster 6 vs. Cluster 0: LLM response (left) and tool call (right).

`compare_cluster_rules`, and `compare_clusters_conformance` are used to compare these clusters. Finally, `get_overall_conformance_stats` is used to get the process baseline. This sequence makes the reasoning chain explicit and ensures that each intermediate conclusion is backed by a specific tool output. Synthesizing these results, the agent finds that while both clusters involve fine issuance, only Cluster 6 includes financial steps like *Payment* and *Add Penalty*. Consequently, deviation patterns differ: Cluster 0 fails on *Send Fine* sequences, whereas Cluster 6 fails on *Payment* sequences, with both falling significantly below the average conformance rate. Despite the complexity of the query, the response remains grounded in the retrieved data, with intermediate conclusions being traceable to specific function call outputs.

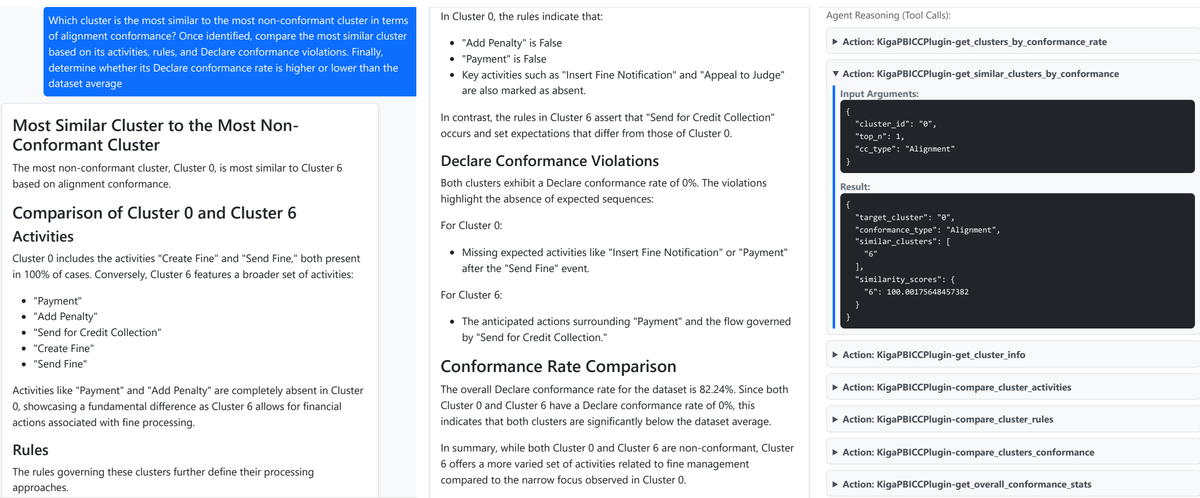

Figure 6: Multi-step analysis: LLM response (left and center) and six chained tool calls (right).

4. Conclusion

Our demonstrator implements a complete pipeline ranging from process discovery to interactive, grounded explanations for clustering and conformance checking. It utilizes hierarchical density-based clustering in combination with the ClusterExplainR to provide clusters and rule-based explanations.

Alignment and declarative conformance checking provide insights into how individual clusters behave in comparison to a normative process model. Finally, an agentic LLM with function calls for retrieval, comparisons, similarity, and context knowledge provides the user with answers that are grounded in factual data, while the possibility to review individual function call results allows users to verify the agent's answers if desired, providing transparent provenance for each claim.

For future work, we plan to develop methods that quantify the contribution of each function call to the generated answer, analogous to agent importance in multi-agent scenarios [26]. We also plan to implement verification mechanisms that ensure every part of a user query is addressed by the appropriate function calls. Likewise, we will investigate approaches to validate that the verbalization accurately reflects the outcomes of the function calls, minimizing misinterpretation and hallucination.

Acknowledgments

Funded by the Bavarian Ministry of Economic Affairs (BayVFP) under KIGA: DIK0313

Declaration on Generative AI

During the preparation of this work, the authors used Gemini, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content. Moreover,the example agentic llm chatbot responses shown in the figures were generated using ChatGPT o3-mini.

References

[1] Y. Liu, S. Pozdniakov, R. Martinez-Maldonado, The effects of visualisation literacy and data storytelling dashboards on teachers' cognitive load, Australasian Journal of Educational Technology (2024). doi:10.14742/ajet.8988.

[2] R. Kosara, J. Mackinlay, Storytelling: The next step for visualization, Computer 46 (2013) 44–50. doi:10.1109/mc.2013.36.

[3] L. Yan, L. Zhao, V. Echeverria, Y. Jin, R. Alfredo, et al., VizChat: Enhancing Learning Analytics Dashboards with Contextualised Explanations Using Multimodal Generative AI Chatbots, Springer Nature Switzerland, 2024, pp. 180–193. doi:10.1007/978-3-031-64299-9_13.

[4] A. Zytek, S. Pidò, K. Veeramachaneni, Llms for xai: Future directions for explaining explanations, 2024. doi:10.48550/ARXIV.2405.06064.

[5] N. Dziri, X. Lu, M. Sclar, X. L. Li, L. Jiang, et al., Faith and fate: Limits of transformers on compositionality, in: A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, et al. (Eds.), Advances in Neural Information Processing Systems, volume 36, Curran Associates, Inc., 2023, pp. 70293–70332. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/deb3c28192f979302c157cb653c15e90-Paper-Conference.pdf.

[6] I. Mirzadeh, K. Alizadeh, H. Shahrokhi, O. Tuzel, S. Bengio, et al., Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models, 2025. doi:10.48550/arXiv.2410.05229. arXiv:2410.05229.

[7] K. Randl, J. Pavlopoulos, A. Henriksson, T. Lindgren, Evaluating the Reliability of Self-explanations in Large Language Models, Springer Nature Switzerland, 2025, pp. 36–51. doi:10.1007/978-3-031-78977-9_3.

[8] R. Manuvinakurike, E. Moss, E. A. Watkins, S. Sahay, G. Raffa, et al., Thoughts without thinking: Reconsidering the explanatory value of chain-of-thought reasoning in llms through agentic pipelines, 2025. doi:10.48550/ARXIV.2505.00875.

[9] P. Spitzer, J. Holstein, K. Morrison, K. Holstein, G. Satzger, et al., Don't be fooled: The misinformation effect of explanations in human–ai collaboration, International Journal of Human–Computer Interaction (2025) 1–29. doi:10.1080/10447318.2025.2574511.

[10] E. Slany, S. Scheele, U. Schmid, Explanatory Interactive Machine Learning with Counterexamples from Constrained Large Language Models, Springer Nature Switzerland, 2024, pp. 324–331. doi:10.1007/978-3-031-70893-0_26.

[11] A. Zytek, S. Pido, S. Alnegheimish, L. Berti-Équille, K. Veeramachaneni, Explingo: Explaining ai predictions using large language models, in: 2024 IEEE International Conference on Big Data (BigData), IEEE, 2024, pp. 1197–1208. doi:10.1109/bigdata62323.2024.10825114.

[12] N. Freyer, H. Kempt, L. Klöser, Easy-read and large language models: on the ethical dimensions of llm-based text simplification, Ethics and Information Technology 26 (2024). doi:10.1007/s10676-024-09792-4.

[13] D. M. Adhikari, A. Hartland, I. Weber, V. K. Cannanure, Exploring llms for automated generation and adaptation of questionnaires, in: Proceedings of the 7th ACM Conference on Conversational User Interfaces, CUI '25, ACM, 2025, pp. 1–23. doi:10.1145/3719160.3736606.

[14] P. Martínez, A. Ramos, L. Moreno, Exploring large language models to generate easy to read content, Frontiers in Computer Science 6 (2024). doi:10.3389/fcomp.2024.1394705.

[15] W. M. P. van der Aalst, Process Mining: A 360 Degree Overview, Springer International Publishing, Cham, 2022, pp. 3–34. doi:10.1007/978-3-031-08848-3_1.

[16] W. van der Aalst, A. Adriansyah, A. K. A. de Medeiros, F. Arcieri, T. Baier, et al., Process mining manifesto, in: F. Daniel, K. Barkaoui, S. Dustdar (Eds.), Business Process Management Workshops, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 169–194.

[17] Z. Wang, C. Huang, X. Yao, A roadmap of explainable artificial intelligence: Explain to whom, when, what and how?, ACM Transactions on Autonomous and Adaptive Systems 19 (2024) 1–40. doi:10.1145/3702004.

[18] J. Amling, E. Slany, C. Dormagen, M. Kretschmann, S. Scheele, Bridging the Interpretability Gap in Process Mining: A Comprehensive Approach Combining Explainable Clustering and Generative AI, Springer Nature Switzerland, 2025, pp. 78–103. doi:10.1007/978-3-032-08324-1_4.

[19] W. van der Aalst, A. Adriansyah, B. van Dongen, Replaying history on process models for conformance checking and performance analysis, WIREs Data Mining and Knowledge Discovery 2 (2012) 182–192. doi:https://doi.org/10.1002/widm.1045.

[20] W. M. P. van der Aalst, M. Pesic, H. Schonenberg, Declarative workflows: Balancing between flexibility and support, Computer Science - Research and Development 23 (2009) 99–113. doi:10.1007/s00450-009-0057-9.

[21] A. Burattin, F. M. Maggi, A. Sperduti, Conformance checking based on multi-perspective declarative process models, Expert Syst. Appl. 65 (2016) 194–211. doi:10.1016/J.ESWA.2016.08.040.

[22] C. Dormagen, J. Amling, S. Scheele, U. Schmid, Explaining process behavior: A declarative framework for interpretable event data, in: Joint Proceedings of the xAI 2025 Late-breaking Work, Demos and Doctoral Consortium co-located with the 3rd World Conference on eXplainable Artificial Intelligence (xAI 2025), Istanbul, Turkey, July 9-11, 2025, volume 4017 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2025, pp. 25–32. URL: https://ceur-ws.org/Vol-4017/paper_04.pdf.

[23] C. L. Werner, J. Amling, C. Dormagen, S. Scheele, Clustxrai: Interactive cluster exploration and explanation for process mining with generative ai, in: Joint Proceedings of the xAI 2025 Late-breaking Work, Demos and Doctoral Consortium (xAI 2025), volume 4017 of *CEUR Workshop Proceedings*, 2025, p. paper 38. URL: https://ceur-ws.org/Vol-4017/paper_38.pdf.

[24] L. McInnes, J. Healy, S. Astels, hdbscan: Hierarchical density based clustering, J. Open Source Softw. 2 (2017) 205. doi:10.21105/JOSS.00205.

[25] J. Amling, S. Scheele, E. Slany, M. Lang, U. Schmid, Explainable AI for Mixed Data Clustering, in: Explainable Artificial Intelligence - 2nd World Conference, xAI 2024, Valletta, Malta, July 17-19, 2024, Proc., Part II, volume 2154 of *Commun. Comput. Inf. Sci.*, Springer, 2024, pp. 42–62. doi:10.1007/978-3-031-63797-1_3.

[26] E. Slany, J. Amling, A. Frummet, M. Lang, S. Scheele, Towards explainable multi-agent systems with graphrag: A guide to explain explanations in the audit domain, in: Explainable AI, Knowledge Representation, and Knowledge Graphs Workshop at European Conference on Artificial Intelligence XAIKRKG@ECAI 2025, to appear.